UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P O Box 1450
Alexandria, Virginia 22313-1450
www uspto gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/589,155	05/04/2007	Claus Pedersen	884A.0148.U1(US)	6810

29683 7590 08/29/2008
HARRINGTON & SMITH, PC
4 RESEARCH DRIVE
SHELTON, CT 06484-6212

EXAMINER	
LEE, CHUN KUAN	
ART UNIT	**PAPER NUMBER**
2181	
MAIL DATE	**DELIVERY MODE**
08/29/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	**Application No.** 10/589,155	**Applicant(s)** PEDERSEN ET AL.	
	Examiner Chun-Kuan Lee	**Art Unit** 2181	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *19 June 2008*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1,3-10,12-28,34-41,43 and 44* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1,3-10,12-28,34-41,43 and 44* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *19 June 2008* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application

6) ☐ Other: _____.

DETAILED ACTION

RESPONSE TO ARGUMENTS

1. Applicant's arguments with respect to claims 1, 3-10, 12-21, 34 and 41 have been considered but are moot in view of the new ground(s) of rejection. Applicant's arguments filed for claims 22-28, 35-40 and 42 have been fully considered but they are not persuasive. Objection to the Drawings is withdrawn. Objection to the abstract is withdrawn. Rejection of claims 12 and 40 under 35 U.S.C.112 second paragraph is withdrawn. Currently, claims 2, 11, 29-33 and 42 are canceled, and claims 1, 3-10, 12-28, 34-41 and 43-44 are pending for examination.

2. In response to applicant's arguments (on page 14) with regard to the independent claim 1 rejected under 35 U.S.C. 103(a) that the Rao teaches "conventional" exe commands specified in the SyncML Representation Protocol; applicant's arguments have fully been considered, but are not found to be persuasive.

The examiner respectfully disagrees, as the examiner does not fully understand as to how the applicant concluded that Rao teaches "conventional" exe commands specified in the SyncML Representation Protocol because Rao's commands are "enhancement commands"; therefore, the commands are enhanced and are not conventional.

3. In response to applicant's arguments (on pages 14-16) regarding the independent claim 1 rejected under 35 U.S.C. 103(a) that the combination of references does not teach/suggest the claimed feature of receiving at an electronic device a command specifying execution of an unidentified executable; applicant's arguments have fully been considered, but are not found to be persuasive.

The examiner respectfully disagrees, as the examiner relied on the combination of references as following for the teaching of the above claimed feature:

Rao teaches receiving at an electronic device (Fig. 1, ref. 107) a command specifying execution of a data (Fig. 1; col. 3, ll. 21-44; col. 6, l. 49 to col. 7, l. 19 and col. 8, l. 25 to col. 12, l. 19), as the update command specifying execution of the firmware update data.

Szeto teach specifying execution of an unidentified executable (e.g. supporting application) on data (Fig. 12A and col. 12, l. 66 to col. 13, l. 16), as the unidentified executable is specified base on the property of the data; more specifically, the received message do not specify the supporting application to be utilized with the message, and after the application type is determined from the received message, the corresponding supporting application is specified.

4. In response to applicant's arguments (on pages 14-17) with regard to the independent claims 22, 24, 35-37 and 40 rejected under 35 U.S.C. 103(a) that the combination of references does not teach/suggest the claimed features determining a content type base on metadata, identifying using content type determined from

metadata, and identifying executable from metadata; applicant's arguments have fully been considered, but are not found to be persuasive.

Please note that the features upon which applicant relies (i.e., determining a content type base on metadata, identifying using content type determined from metadata, and identifying executable from metadata) are not recited in the rejected claim(s). Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

I. <u>OBJECTION TO THE CLAIMS</u>

5. Claim 43 is objected to because of the following informalities:

in claim 43, line 3, "hierarchiacal" should be replace with -hierarchical-.

Appropriate correction is required.

II. <u>REJECTIONS BASED ON 35 U.S.C. 112</u>

Claim Rejections - 35 USC § 112

The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

6. Claims 16, 21, 34 and 43-44 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

As per claim 16, in line 2, it is not fully clear if "an unidentified executable" is the same/different unidentified executable previously recited; the examiner will assume the claimed limitation of “the unidentified executable” for the current examination.

As per claim 21, in line 4, it is not fully clear if “a content type” is the same/different content type previously recited; the examiner will assume the claimed limitation of “the content type” for the current examination.

As per claim 34, in line 4, it is not fully clear if “first data” is the same/different first data previously recited; the examiner will assume the claimed limitation of “the first data” for the current examination.

As per claim 43, in line 8, it is not fully clear if “a content type” is the same/different content type previously recited; the examiner will assume the claimed limitation of “the content type” for the current examination.

As per claim 44, in line 9, it is not fully clear if “a content type” is the same/different content type previously recited; the examiner will assume the claimed limitation of “the content type” for the current examination.

III. <u>REJECTIONS BASED ON PRIOR ART</u>

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

7. Claims 1-21, 34, 41 and 43-44 are rejected under 35 U.S.C. 103(a) as being unpatentable over Rao et al. (US Patent 6,978,453) in view of "SyncML Meta-Information DTD" and Szeto (US Patent 7,188,143).

8. As per claim 1, Rao teaches a method comprising:

receiving at an electronic device (Fig. 1, ref. 107) a command (e.g. update command) specifying execution on first data (Fig. 1; col. 3, ll. 21-44; col. 6, l. 49 to col. 7, l. 19 and col. 8, l. 25 to col. 12, l. 19), wherein the first data executed to be associated with firmware update data;

utilization of metadata protocol, wherein the first leaf node would have the corresponding metadata (col. 6, l. 49 to col. 7, l. 19);

automatically determining (e.g. determining via recognition) a property of the identified first data (e.g. property identifying first data to be firmware update data) (col. 8, l. 25 to col. 12, l. 19), as the received command is recognized by the electronic device to have the property associated with firmware updating;

operating on the identified first data using an executable (e.g. module)(col. 5, ll. 23-32 and col. 5, l. 61 to col. 6, l. 4), as the module would operate on the firmware update data via downloading and updating processes.

Rao does not teach the method comprising: an unidentified executable; determine content type from the metadata; and automatically identifying an executable using the content type determined from the metadata.

SyncML Meta-Information DTD" teaches the metadata indicating a content type (Sec. 3-5 on pp. 5-12), as it is well known that metadata is data about data and SyncML have meta-information such as parameter or attributes that are about type or content of data; therefore, metadata may be utilized for determining the content type of data.

Szeto teach a system and method comprising:

an unidentified executable (e.g. unidentified supporting application) (Fig. 12A and col. 12, l. 66 to col. 13, l. 16), as the received message do not identify the supporting application to be utilized with the message, and after the application type is determined from the received message, the corresponding supporting application is identified;

determine a content type (e.g. application type) from metadata (Fig. 12A and col. 12, l. 66 to col. 13, l. 16), in combination with SyncML Meta-Information DTD's teaching, the received message having metadata with content types including movie trailer, game, animation cartoon, advertisement, and flash presentation, and to determine the supporting application for the received message, the metadata is examining to determine the application type (e.g. content type); and

automatically identifying an executable (e.g. supporting application) using the content type determined from metadata (Fig. 12A and col. 12, l. 66 to col. 13, l. 16), in combination with SyncML Meta-Information DTD's teaching, the received message would be associated with content types including movie trailer, game, animation cartoon, advertisement, and flash presentation in the metadata, and by using the content type, the corresponding supporting application would be identified.

It would have been obvious for one of ordinary skill in this art, at the time of invention was made to include SyncML Meta-Information DTD's content type and metadata and Szeto's identification of the executable into Rao's operation on the first identified data for the benefit of properly operating in accordance SyncML standard as in Rao's system and also for the benefit to the having a reliable system and method for a user to execute and control application (Szeto, col. 2, ll. 30-33) to obtain the invention as specified in claim 1.

9. As per claim 3, Rao, SyncML Meta-Information DTD and Szeto teach all the limitations of claim 1 as discussed above, where Rao further teaches the method comprising wherein the command contains an identifier (e.g. URI) of the first data (Rao, col. 6, l. 49 to col. 7, l. 19 and col. 8, ll. 25-34).

10. As per claim 4, Rao, SyncML Meta-Information DTD and Szeto teach all the limitations of claim 3 as discussed above, where Rao further teaches the method comprising wherein the identifier identifies a node of a hierarchical nodular data structure (e.g. tree data structure) (Rao, col. 6, l. 49 to col. 7, l. 19 and col. 8, ll. 25-34).

11. As per claim 5, Rao, SyncML Meta-Information DTD and Szeto teach all the limitations of claim 4 as discussed above, where Rao further teaches the method comprising wherein the command is an exec command and the identifier is a URI

contained within a source element, which is contained within the exec command (Rao, col. 6, l. 49 to col. 7, l. 19 and col. 8, l. 25 to col. 12, l. 19).

12. As per claim 6, Rao, SyncML Meta-Information DTD and Szeto teach all the limitations of claim 1 as discussed above, where Rao and Szeto further teach the method comprising wherein the command is received as XML code (Rao, col. 6, ll. 49 to col. 7, l. 3 and Szeto, col. 7, ll. 48-53).

13. As per claim 7, Rao, SyncML Meta-Information DTD and Szeto teach all the limitations of claim 6 as discussed above, where Rao further teaches the method comprising wherein the command is a SyncML command (Rao, col. 6, ll. 49 to col. 7, l. 3 and col. 8, l. 25 to col. 12, l. 19).

14. As per claim 8, Rao, SyncML Meta-Information DTD and Szeto teach all the limitations of claim 1 as discussed above, where Rao further teaches the method comprising wherein the identified first data is stored at the electronic device (Rao, col. 3, ll. 52-63; col. 5, ll. 23-32; col. 7, ll. 38-41 and col. 11, l. 48 to col. 12, l. 19).

15. As per claim 9, Rao, SyncML Meta-Information DTD and Szeto teach all the limitations of claim 6 as discussed above, where Rao further teaches the method comprising wherein the identified first data is stored at a first leaf node of a hierarchical nodular data structure (e.g. tree data structure) (Rao, col. 3, ll. 52-63; col. 6, l. 49 to col.

7, l. 19; col. 8, ll. 25-34 and col. 11, l. 48 to col. 12, l. 19), as the data would be store in the first leaf node of the tree data structure.

16. As per claim 10, Rao, SyncML Meta-Information DTD and Szeto teach all the limitations of claim 9 as discussed above, where all further teach the method comprising wherein the metadata is associated with the first leaf node and identifies the content type of the first data stored at the first leaf node of the hierarchical data structure (e.g. tree data structure) (Rao, col. 6, l. 49 to col. 7, l. 19; col. 8, l. 25 to col. 12, l. 19, SyncML Meta-Information DTD, pp. 5-6, and Szeto, Fig. 12A; col. 12, l. 66 to col. 13, l. 16).

17. As per claim 12, Rao, SyncML Meta-Information DTD and Szeto teach all the limitations of claim 1 as discussed above, where SyncML Meta-Information DTD and Szeto further teach the method comprising wherein determining the content type uses at least one of the value of a Format element and the value of a Type element associated with the first data (SyncML Meta-Information DTD, pp. 5-12 and Szeto, Fig. 12A and col. 12, l. 66 to col. 13, l. 16).

18. As per claim 13, Rao, SyncML Meta-Information DTD and Szeto teach all the limitations of claim 1 as discussed above, where Szeto further teaches the method comprising associating a plurality of different executables (e.g. different supporting applications for movie trailer, game, animation cartoon, advertisement, flash presentation) with each of a plurality of different content types (Szeto, Fig. 12A and col.

12, l. 66 to col. 13, l. 16), as each different content types have the corresponding supporting application.

19. As per claim 14, Rao, SyncML Meta-Information DTD and Szeto teach all the limitations of claim 13 as discussed above, where SyncML Meta-Information DTD and Szeto further teach the method comprising wherein automatically identifying an executable from the content type comprises identifying the executable associated with the content type (SyncML Meta-Information DTD, pp. 5-12 and Szeto, Fig. 12A and col. 12, l. 66 to col. 13, l. 16).

20. As per claim 15, Rao, SyncML Meta-Information DTD and Szeto teach all the limitations of claim 13 as discussed above, where Rao and Szeto further teach the method comprising wherein the plurality of different executables are stored in the electronic device (Rao, Fig 1; col. 5, l. 23 to col. 6, l. 4 and Szeto, Fig. 12A; col. 12, l. 66 to col. 13, l. 16), as the electronic devise would have the corresponding supporting application for operating the first data.

21. As per claim 16, Rao, SyncML Meta-Information DTD and Szeto teach all the limitations of claim 1 as discussed above, where Rao further teaches the method comprising before receiving the command specifying execution of the unidentified executable on the first data, receiving commands for creating a hierarchical nodular data structure (e.g. tree data structure) including the first data at the electronic device

(Rao, col. 6, l. 49 to col. 7, l. 19 and col. 7, ll. 38-41 SyncML Meta-Information DTD, pp. 5-12), as the tree data structure is created prior to the execution of the update command.

22. As per claim 17, Rao teaches a method comprising:

transferring code comprising a command to an electronic device (Fig. 1, ref. 107), wherein the command specifies execution on first data stored at a first leaf node of a hierarchical nodular data structure (e.g. tree data structure) (Fig. 1; col. 3, ll. 21-44; col. 6, l. 49 to col. 7, l. 19 and col. 8, l. 25 to col. 12, l. 19), as the first leaf node would reside within the tree data structure;

utilization of metadata protocol, wherein the first leaf node would have the corresponding metadata (col. 6, l. 49 to col. 7, l. 19);

determining (e.g. determining via recognition) a property of the identified first data (e.g. property identifying first data to be firmware update data) (col. 8, l. 25 to col. 12, l. 19), as the received command is recognized by the electronic device to have the property associated with firmware updating; and

operating on the first data store at the identified first leaf node using an executable (e.g. module)(col. 5, ll. 23-32; col. 5, l. 61 to col. 6, l. 4 and col. 8, l. 25 to col. 12, l. 19), as the module would operate on the firmware update data via downloading and updating processes.

Rao does not teach the method comprising: an unidentified executable; determine a content type from the metadata; and identifying an executable using the content type determined from the metadata for operation.

SyncML Meta-Information DTD" teaches the metadata indicating a content type (Sec. 3-5 on pp. 5-12), as it is well known that metadata is data about data and SyncML have meta-information such as parameter or attributes that are about type or content of data; therefore, metadata may be utilized for determining the content type of data.

Szeto teach a system and method comprising:

an unidentified executable (e.g. unidentified supporting application) (Fig. 12A and col. 12, l. 66 to col. 13, l. 16), as the received message do not identify the supporting application to be utilized with the message, and after the application type is determined from the received message, the corresponding supporting application is identified;

determine a content type (e.g. application type) from metadata (Fig. 12A and col. 12, l. 66 to col. 13, l. 16), in combination with SyncML Meta-Information DTD's teaching, the received message having metadata with content types including movie trailer, game, animation cartoon, advertisement, and flash presentation, and to determine the supporting application for the received message, the metadata is examining to determine the application type (e.g. content type); and

identifying an executable (e.g. supporting application) using the content type determined from metadata for operation (Fig. 12A and col. 12, l. 66 to col. 13, l. 16), in combination with SyncML Meta-Information DTD's teaching, the received message would be associated with content types including movie trailer, game, animation cartoon, advertisement, and flash presentation in the metadata, and by using the content type, the corresponding supporting application would be identified.

It would have been obvious for one of ordinary skill in this art, at the time of invention was made to include SyncML Meta-Information DTD's content type and metadata and Szeto's identification of the executable into Rao's operation on the first identified data for the benefit of properly operating in accordance SyncML standard as in Rao's system and also for the benefit to the having a reliable system and method for a user to execute and control application (Szeto, col. 2, ll. 30-33) to obtain the invention as specified in claim 17.

23. As per claim 18, Rao teaches a method, comprising:

receiving re-usable code at an electronic device (Fig. 1, ref. 107) (Fig. 1; col. 2, ll. 3-20; col. 3, ll. 21-44; col. 6, l. 49 to col. 7, l. 19 and col. 8, l. 25 to col. 12, l. 19) (e.g. the code would be re-usable as SyncML specification enable operation with any mobile device) wherein the code comprises:

commands for creating at the electronic device a hierarchical nodular data structure (e.g. tree data structure), having leaf nodes and interior nodes (i.e. the tree data structure would have the corresponding leaf nodes and interior nodes), that comprises first data stored at a first leaf node; and a further command specifying execution on the first data stored at the first leaf node (e.g. executing firmware update data at first leaf node) (Fig. 1; col. 3, ll. 21-44; col. 6, l. 49 to col. 7, l. 19; col. 7, ll. 38-41 and col. 8, l. 25 to col. 12, l. 19);

utilization of metadata protocol, wherein the first leaf node would have the corresponding metadata (col. 6, l. 49 to col. 7, l. 19);

determining (e.g. determining via recognition) a property of the first data stored at the first leaf node (col. 8, l. 25 to col. 12, l. 19), as the received command is recognized by the electronic device to have the property associated with firmware update data;

operating on the first data, stored at the first leaf node, using an executable (e.g. module) (col. 5, ll. 23-32 and col. 5, l. 61 to col. 6, l. 4), as the module would operate on the firmware update data via downloading and updating processes.

Rao does not teach the method comprising: an unidentified executable; determine a content type from the metadata; and identifying an executable using the content type determined from the metadata for operation.

SyncML Meta-Information DTD" teaches the metadata indicating a content type (Sec. 3-5 on pp. 5-12), as it is well known that metadata is data about data and SyncML have meta-information such as parameter or attributes that are about type or content of data; therefore, metadata may be utilized for determining the content type of data.

Szeto teach a system and method comprising:

an unidentified executable (e.g. unidentified supporting application) (Fig. 12A and col. 12, l. 66 to col. 13, l. 16), as the received message do not identify the supporting application to be utilized with the message, and after the application type is determined from the received message, the corresponding supporting application is identified;

determine a content type (e.g. application type) from metadata (Fig. 12A and col. 12, l. 66 to col. 13, l. 16), in combination with SyncML Meta-Information DTD's teaching, the received message having metadata with content types including movie trailer, game, animation cartoon, advertisement, and flash presentation, and to determine the supporting application for the received message, the metadata is examining to determine the application type (e.g. content type); and

identifying an executable (e.g. supporting application) using the content type determined from metadata for operation (Fig. 12A and col. 12, l. 66 to col. 13, l. 16), in combination with SyncML Meta-Information DTD's teaching, the received message would be associated with content types including movie trailer, game, animation cartoon, advertisement, and flash presentation in the metadata, and by using the content type, the corresponding supporting application would be identified.

It would have been obvious for one of ordinary skill in this art, at the time of invention was made to include SyncML Meta-Information DTD's content type and metadata and Szeto's identification of the executable into Rao's operation on the first identified data for the benefit of properly operating in accordance SyncML standard as in Rao's system and also for the benefit to the having a reliable system and method for a user to execute and control application (Szeto, col. 2, ll. 30-33) to obtain the invention as specified in claim 18.

24. As per claim 19, Rao teaches an electronic device, comprising:

a memory configure to store first data and metadata for first data (Fig. 1; col. 3, ll. 21-63; col. 6, l. 49 to col. 7, l. 19 and col. 8, l. 25 to col. 12, l. 19), as metadata protocol is utilized;

a receiver configured to receive a command specifying execution of the first data (Fig. 1; col. 3, ll. 21-44; col. 6, l. 49 to col. 7, l. 19 and col. 8, l. 25 to col. 12, l. 19), as the mobile device (Fig. 1, ref. 107) must have the receiver in order to receive commands from the SyncML server (Fig. 1, ref. 109) for execution of firmware updating; and

a processor (Fig. 1, ref. 107) configured to determine a property of the first data and operate on the data using an executable (col. 5, ll. 23-32; col. 5, l. 61 to col. 6, l. 4 and col. 8, l. 25 to col. 12, l. 19), as the received command is recognized by the electronic device to have the property associated with firmware updating, which the module would then operate on the firmware update data via downloading and updating processes.

Rao does not teach the method comprising: an unidentified executable; determine a content type from the metadata; and identifying an executable using the content type determined from the metadata for operation.

SyncML Meta-Information DTD" teaches the metadata indicating a content type (Sec. 3-5 on pp. 5-12), as it is well known that metadata is data about data and SyncML have meta-information such as parameter or attributes that are about type or content of data; therefore, metadata may be utilized for determining the content type of data.

Szeto teach a system and method comprising:

an unidentified executable (e.g. unidentified supporting application) (Fig. 12A and col. 12, l. 66 to col. 13, l. 16), as the received message do not identify the supporting application to be utilized with the message, and after the application type is determined from the received message, the corresponding supporting application is identified;

determine a content type (e.g. application type) from metadata (Fig. 12A and col. 12, l. 66 to col. 13, l. 16), in combination with SyncML Meta-Information DTD's teaching, the received message having metadata with content types including movie trailer, game, animation cartoon, advertisement, and flash presentation, and to determine the supporting application for the received message, the metadata is examining to determine the application type (e.g. content type); and

identifying an executable (e.g. supporting application) using the content type determined from metadata for operation (Fig. 12A and col. 12, l. 66 to col. 13, l. 16), in combination with SyncML Meta-Information DTD's teaching, the received message would be associated with content types including movie trailer, game, animation cartoon, advertisement, and flash presentation in the metadata, and by using the content type, the corresponding supporting application would be identified.

It would have been obvious for one of ordinary skill in this art, at the time of invention was made to include SyncML Meta-Information DTD's content type and metadata and Szeto's identification of the executable into Rao's operation on the first identified data for the benefit of properly operating in accordance SyncML standard as in Rao's system and also for the benefit to the having a reliable system and method for a user to execute and control application (Szeto, col. 2, ll. 30-33) to obtain the invention as specified in claim 19.

25. As per claim 20, Rao, SyncML Meta-Information DTD and Szeto teach all the limitations of claim 19 as discussed above, where Rao further teaches the electronic

device comprising wherein the receiver is further configured to receive a set-up code (e.g. set-up code such as add command), and the processor is operable to interpret the received set-up code to create a hierarchical nodular data structure (e.g. tree data structure), having leaf nodes and interior nodes, that comprises a first leaf node storing the first data (Rao, Fig. 1; col. 3, ll. 21-44; col. 6, l. 49 to col. 7, l. 19; col. 7, ll. 38-41 and col. 8, l. 25 to col. 12, l. 19), as the created tree data structure have the corresponding leaf nodes and interior nodes.

26. As per claim 21, Rao, SyncML Meta-Information DTD and Szeto teach all the limitations of claim 20 as discussed above, where all further teach the electronic device comprising wherein the receiver is configured to received the command in the set up code, and the processor is configured to interpret the command to determine, from the metadata of the first data, the content type (e.g. firmware update, movie trailer, game, animation cartoon, advertisement, flash presentation) of the first data (Rao, Fig. 1; col. 3, ll. 21-44; col. 5, ll. 23-32; col. 5, l. 61 to col. 6, l. 4; col. 6, l. 49 to col. 7, l. 19; col. 8, l. 25 to col. 12, l. 19; SyncML Meta-Information DTD, Sec. 3-5 on pp. 5-12, and Szeto, Fig. 12A; col. 12, l. 66 to col. 13, l. 16).

27. As per claim 34, Rao an electronic device (Fig. 1, ref. 107), comprising:

means for storing first data (Fig. 1; col. 3, ll. 52-63; col. 6, l. 49 to col. 7, l. 19 and col. 8, l. 25 to col. 12, l. 19);

means for receiving a command specifying execution on the first data (Fig. 1; col. 3, ll. 21-44; col. 6, l. 49 to col. 7, l. 19 and col. 8, l. 25 to col. 12, l. 19), wherein the first data is identified to be associated with firmware update data;

utilization of metadata protocol, wherein the first leaf node would have the corresponding metadata (col. 6, l. 49 to col. 7, l. 19);

means for determining (e.g. determining via recognition) a property of the identified first data (e.g. property identifying first data to be firmware update data) (col. 8, l. 25 to col. 12, l. 19), as the received command is recognized by the electronic device to have the property associated with firmware updating;

means for operating on the identified data using an executable (e.g. module)(col. 5, ll. 23-32 and col. 5, l. 61 to col. 6, l. 4), as the module would operate on the firmware update data via downloading and updating processes.

Rao does not teach the method comprising: an unidentified executable; determine a content type from the metadata; and means for identifying an executable using the content type determined from the metadata for operating.

SyncML Meta-Information DTD" teaches the metadata indicating a content type (Sec. 3-5 on pp. 5-12), as it is well known that metadata is data about data and SyncML have meta-information such as parameter or attributes that are about type or content of data; therefore, metadata may be utilized for determining the content type of data.

Szeto teach a system and method comprising:

an unidentified executable (e.g. unidentified supporting application) (Fig. 12A and col. 12, l. 66 to col. 13, l. 16), as the received message do not identify the supporting application to be utilized with the message, and after the application type is determined from the received message, the corresponding supporting application is identified;

determine a content type (e.g. application type) from metadata (Fig. 12A and col. 12, l. 66 to col. 13, l. 16), in combination with SyncML Meta-Information DTD's teaching, the received message having metadata with content types including movie trailer, game, animation cartoon, advertisement, and flash presentation, and to determine the supporting application for the received message, the metadata is examining to determine the application type (e.g. content type); and

means for identifying an executable (e.g. supporting application) using the content type determined from metadata for operation (Fig. 12A and col. 12, l. 66 to col. 13, l. 16), in combination with SyncML Meta-Information DTD's teaching, the received message would be associated with content types including movie trailer, game, animation cartoon, advertisement, and flash presentation in the metadata, and by using the content type, the corresponding supporting application would be identified.

It would have been obvious for one of ordinary skill in this art, at the time of invention was made to include SyncML Meta-Information DTD's content type and metadata and Szeto's identification of the executable into Rao's operation on the first identified data for the benefit of properly operating in accordance SyncML standard as in Rao's system and also for the benefit to the having a reliable system and method for a user to execute and control application (Szeto, col. 2, ll. 30-33) to obtain the invention as specified in claim 34.

28. As per claim 41, Rao teaches a computer program product comprising program instructions embodied on a tangible computer readable-readable medium, execution of the program instructions resulting in operations comprising:

automatically determining (e.g. determining via recognition) a property of a first data (e.g. property identifying first data to be firmware update data) (Fig. 1; col. 3, ll. 21-44; col. 6, l. 49 to col. 7, l. 19 and col. 8, l. 25 to col. 12, l. 19), as the received command is recognized by the electronic device to have the property associated with firmware updating;

utilization of metadata protocol, wherein the first data would have the corresponding metadata (col. 6, l. 49 to col. 7, l. 19); and

enabling the first data to be operated on using an executable (e.g. module)(col. 5, ll. 23-32 and col. 5, l. 61 to col. 6, l. 4), as the module would operate on the firmware update data via downloading and updating processes.

Rao does not teach the computer program product comprising: determine a content type from the metadata; and automatically identifying an executable using the content type determined from the metadata for operation.

SyncML Meta-Information DTD" teaches the metadata indicating a content type (Sec. 3-5 on pp. 5-12), as it is well known that metadata is data about data and SyncML have meta-information such as parameter or attributes that are about type or content of data; therefore, metadata may be utilized for determining the content type of data.

Szeto teach a system and method comprising:

determine a content type (e.g. application type) from metadata (Fig. 12A and col. 12, l. 66 to col. 13, l. 16), in combination with SyncML Meta-Information DTD's teaching, the received message having metadata with content types including movie trailer, game, animation cartoon, advertisement, and flash presentation, and to determine the supporting application for the received message, the metadata is examining to determine the application type (e.g. content type); and

automatic identifying an executable (e.g. supporting application) using the content type determined from metadata for operation (Fig. 12A and col. 12, l. 66 to col. 13, l. 16), in combination with SyncML Meta-Information DTD's teaching, the received message would be associated with content types including movie trailer, game, animation cartoon, advertisement, and flash presentation in the metadata, and by using the content type, the corresponding supporting application would be identified.

It would have been obvious for one of ordinary skill in this art, at the time of invention was made to include Szeto's identification of the executable into Rao's operation of the first data for the benefit of having a reliable system and method for a user to execute and control application (Szeto, col. 2, ll. 30-33) to obtain the invention as specified in claim 41.

29. As per claims 43-44, Rao teaches an electronic device and a method comprising:

a receiver configured to receive a first command at an electronic device (Fig. 1, ref. 107), the first command specifying creation of a leaf node in a hierarchical data structure (e.g. tree data structure with the corresponding leaf nodes), and identifying first data (e.g. firmware updating data) to be stored at the leaf node and metadata of the first data (Fig. 1; col. 3, ll. 21-44; col. 6, l. 49 to col. 7, l. 41 and col. 8, l. 25 to col. 12, l. 19), wherein the mobile device (Fig. 1, ref. 107) must have the receiver in order to receive command from the SyncML server (Fig. 1, ref. 109); and

a processor configured to create the leaf node at the electronic device (Fig. 1; col. 3, ll. 21-44; col. 6, l. 49 to col. 7, l. 41 and col. 8, l. 25 to col. 12, l. 19), wherein

the receiver is further configured to receive a second command, at the electronic device, that specifies execution on the first data stored at the created leaf node, and the processor is further configured to operate on the first data using an executable (Fig. 1; col. 3, ll. 21-44; col. 5, l. 61 to col. 6, l. 4; col. 6, l. 49 to col. 7, l. 41 and col. 8, l. 25 to col. 12, l. 19).

Rao does not teach the electronic device and the method comprising: the metadata indicating a content type; an unidentified executable; determination of the content type from the metadata; and identify an executable using the content type determined from the metadata for operation.

SyncML Meta-Information DTD" teaches the metadata indicating a content type (Sec. 3-5 on pp. 5-12), as it is well known that metadata is data about data and SyncML have meta-information such as parameter or attributes that are about type or content of data; therefore, metadata may be utilized for determining the content type of data.

Szeto teach a system and method comprising:

an unidentified executable (e.g. unidentified supporting application) (Fig. 12A and col. 12, l. 66 to col. 13, l. 16), as the received message do not identify the supporting application to be utilized with the message, and after the application type is determined from the received message, the corresponding supporting application is identified;

determination of the content type (e.g. application type) from the metadata (Fig. 12A and col. 12, l. 66 to col. 13, l. 16), in combination with SyncML Meta-Information DTD's teaching, the received message having metadata with content types including movie trailer, game, animation cartoon, advertisement, and flash presentation, and to determine the supporting application for the received message, the metadata is examining to determine the application type (e.g. content type); and

identifying an executable (e.g. supporting application) using the content type determined from metadata for operation (Fig. 12A and col. 12, l. 66 to col. 13, l. 16), in combination with SyncML Meta-Information DTD's teaching, the received metadata would be associated with content types including movie trailer, game, animation cartoon, advertisement, and flash presentation in the metadata, and by using the content type, the corresponding supporting application would be identified.

It would have been obvious for one of ordinary skill in this art, at the time of invention was made to include SyncML Meta-Information DTD's content type and metadata and Szeto's identification of the executable into Rao's operation on the first identified data for the benefit of properly operating in accordance SyncML standard as in Rao's system and also for the benefit to the having a reliable system and method for a user to execute and control application (Szeto, col. 2, ll. 30-33) to obtain the invention as specified in claims 43-44.

30. Claims 22-28 and 35-40 are rejected under 35 U.S.C. 103(a) as being unpatentable over Rao et al. (US Patent 6,978,453) in view of Szeto (US Patent 7,188,143).

31. As per claim 22, Rao teaches a data structure embodied on a computer readable medium, comprising: code identifying first data (Fig. 1; col. 3, ll. 21-44; col. 6, l. 49 to col. 7, l. 19 and col. 8, l. 25 to col. 12, l. 19), as the first data associated with the firmware update data is identified; and

execution on the first data (col. 5, ll. 23-32 and col. 5, l. 61 to col. 6, l. 4), as the module would execute on the first data via downloading and updating processes.

Rao does not teach the data structure comprising specifying execution of an unidentified executable on the first data.

Szeto teach a system and method comprising specifying execution of an unidentified executable (e.g. supporting application) on a first data (Fig. 12A and col. 12, l. 66 to col. 13, l. 16), as the unidentified executable is specified base on the property of the first data; more specifically, the received message do not specify the supporting application to be utilized with the message, and after the application type is determined from the received message, the corresponding supporting application is specified.

It would have been obvious for one of ordinary skill in this art, at the time of invention was made to include Szeto's specification of the unidentified executable into Rao's execution of the first data for the benefit of having a reliable system and method for a user to execute and control application (Szeto, col. 2, ll. 30-33) to obtain the invention as specified in claim 22.

32. As per claim 23, Rao and Szeto teach all the limitations of claim 22 as discussed above, where Rao further teaches the data structure comprising wherein the code further specifies the transfer of the first data to an electronic device (Rao, Fig. 1, ref. 107) (Rao, Fig. 1; col. 6, l. 49 to col. 7, l. 19 and col. 8, l. 25 to col. 12, l. 19).

33. As per claim 24, Rao teaches a data structure embodied on a computer readable medium, comprising:

commands, execution of which create at an electronic device (Fig. 1, ref. 107) a hierarchical nodular data structure (e.g. tree data structure), having leaf nodes and interior nodes, that comprises first data stored at a first leaf node (Fig. 1; col. 3, ll. 21-44; col. 6, l. 49 to col. 7, l. 19; col. 7, ll. 38-41 and col. 8, l. 25 to col. 12, l. 19), as the tree data structure would have the leaf nodes and interior nodes; and

a further command identifying the first leaf node that an executable (e.g. module) would operate on the first data stored at the first leaf node (col. 5, ll. 23-32; col. 5, l. 61 to col. 6, l. 4 and col. 8, l. 25 to col. 12, l. 19), as the module would operate on the firmware update data via downloading and updating processes.

Rao does not teach the data structure comprising specifying execution of an unidentified executable on the first data.

Szeto teach a system and method comprising specifying execution of an unidentified executable (e.g. supporting application) on a first data (Fig. 12A and col. 12, l. 66 to col. 13, l. 16), as the unidentified executable is specified base on the property of the first data; more specifically, the received message do not specify the supporting application to be utilized with the message, and after the application type is determined from the received message, the corresponding supporting application is specified.

It would have been obvious for one of ordinary skill in this art, at the time of invention was made to include Szeto's specification of the unidentified executable into Rao's execution of the first data for the benefit of having a reliable system and method for a user to execute and control application (Szeto, col. 2, ll. 30-33) to obtain the invention as specified in claim 24.

34. As per claim 25, Rao and Szeto teach all the limitations of claim 22 as discussed above, where both further teach a method, comprising: using a data structure as claimed in claim 22 (Rao, Fig. 1; col. 3, ll. 21-44; col. 6, l. 49 to col. 7, l. 19; col. 7, ll. 38-41 and col. 8, l. 25 to col. 12, l. 19 and Szeto, Fig. 12A and col. 12, l. 66 to col. 13, l. 16).

35. As per claim 26, Rao and Szeto teach all the limitations of claim 22 as discussed above, where both further teach a method comprising: setting-up an electronic device (Rao, Fig. 1, ref. 107) using a data structure as claimed in claim 22 (Rao, Fig. 1; col. 3,

ll. 21-44; col. 6, l. 49 to col. 7, l. 19; col. 7, ll. 38-41 and col. 8, l. 25 to col. 12, l. 19 and Szeto, Fig. 12A and col. 12, l. 66 to col. 13, l. 16).

36. As per claim 27, Rao and Szeto teach all the limitations of claim 22 as discussed above, where both further teach a method comprising: re-using the data structure as claimed in claim 22, to set-up different electronic devices (Rao, Fig. 1; col. 2, ll. 3-20; col. 3, ll. 21-44; col. 6, l. 49 to col. 7, l. 19; col. 7, ll. 38-41 and col. 8, l. 25 to col. 12, l. 19 and Szeto, Fig. 12A and col. 12, l. 66 to col. 13, l. 16), as the SyncML specification is able to work with any mobile device, therefore, the data structure would be re-usable.

37. As per claim 28, Rao and Szeto teach all the limitations of claim 22 as discussed above, where Rao further teaches a server (Rao, Fig. 1, ref. 109) for storing and transmitting the data structure as claimed in claim 22 (Rao, Fig. 1; col. 3, ll. 21-44; col. 6, l. 49 to col. 7, l. 19; col. 7, ll. 38-41 and col. 8, l. 25 to col. 12, l. 19).

38. As per claim 35, Rao teaches a method, comprising: providing code identifying first data (e.g. firmware update data); and transmitting the code (e.g. transmit to the mobile handset 107 of Fig 1) (Fig. 1; col. 3, ll. 21-44; col. 6, l. 49 to col. 7, l. 19 and col. 8, l. 25 to col. 12, l. 19).

Rao does not teach the method comprising specifying execution of an unidentified executable on the first data.

Szeto teach a system and method comprising specifying execution of an unidentified executable (e.g. supporting application) on a first data (Fig. 12A and col. 12, l. 66 to col. 13, l. 16), as the specification of the supporting application is in accordance with the property of the first data; more specifically, the received message do not specify the supporting application to be utilized with the message, and after the application type is determined from the received message, the corresponding supporting application is specified.

It would have been obvious for one of ordinary skill in this art, at the time of invention was made to include Szeto's specification of the unidentified executable into Rao's operation of the first data for the benefit of having a reliable system and method for a user to execute and control application (Szeto, col. 2, ll. 30-33) to obtain the invention as specified in claim 35.

39. As per claim 36, Rao teaches a method, comprising:

transmitting commands for creating a hierarchical nodular data structure (e.g. tree data structure), having leaf nodes and interior nodes, that comprises first data stored at a first leaf node (Fig. 1; col. 3, ll. 21-44; col. 6, l. 49 to col. 7, l. 19; col. 7, ll. 38-41 and col. 8, l. 25 to col. 12, l. 19), as the tree data structure would have the leaf nodes and interior nodes; and

transmitting a further command specifying execution on the first data stored at the first leaf node (col. 5, ll. 23-32; col. 5, l. 61 to col. 6, l. 4 and col. 8, l. 25 to col. 12, l.

19), as the module would operate on the firmware update data via downloading and updating processes.

Rao does not teach the method comprising specifying execution of an unidentified executable on the first data.

Szeto teach a system and method comprising specifying execution of an unidentified executable (e.g. supporting application) on a first data (Fig. 12A and col. 12, l. 66 to col. 13, l. 16), as the specification of the supporting application is in accordance with the property of the first data; more specifically, the received message do not specify the supporting application to be utilized with the message, and after the application type is determined from the received message, the corresponding supporting application is specified.

It would have been obvious for one of ordinary skill in this art, at the time of invention was made to include Szeto's specification of the unidentified executable into Rao's operation of the first data for the benefit of having a reliable system and method for a user to execute and control application (Szeto, col. 2, ll. 30-33) to obtain the invention as specified in claim 36.

40. As per claim 37, Rao teaches a server (Fig. 1, ref. 109), comprising:

a memory configured to store a code identifying first data and operating on the first data; and an interface configure to transmit the code (Fig. 1; col. 3, ll. 21-44; col. 4, l. 62 to col. 5, l. 8; col. 6, l. 49 to col. 7, l. 19; col. 7, ll. 38-41 and col. 8, l. 25 to col. 12, l. 19), wherein the interface may includes the SyncML engine (Fig. 1, ref. 137).

Rao does not teach the server comprising specifying execution of an unidentified executable on the first data.

Szeto teach a system and method comprising specifying execution of an unidentified executable (e.g. supporting application) on a first data (Fig. 12A and col. 12, l. 66 to col. 13, l. 16), as the specification of the supporting application is in accordance with the property of the first data; more specifically, the received message do not specify the supporting application to be utilized with the message, and after the application type is determined from the received message, the corresponding supporting application is specified.

It would have been obvious for one of ordinary skill in this art, at the time of invention was made to include Szeto's specification of the unidentified executable into Rao's operation of the first data for the benefit of having a reliable system and method for a user to execute and control application (Szeto, col. 2, ll. 30-33) to obtain the invention as specified in claim 37.

41. As per claim 38, Rao and Szeto teach all the limitations of claim 37 as discussed above, where Rao further teaches a server comprising wherein the operations further comprise setting up an electronic device (Rao, Fig. 1, ref. 107) (Rao, Fig. 1; col. 3, ll. 21-44; l. 8; col. 6, l. 49 to col. 7, l. 19; col. 7, ll. 38-41 and col. 8, l. 25 to col. 12, l. 19), as the firmware update data is utilized for setting up the mobile handset.

42. As per claim 39, Rao and Szeto teach all the limitations of claim 37 as discussed above, where Rao further teaches a server comprising wherein the operations further comprise re-using the code in setting up different electronic devices (Rao, Fig. 1; col. 2, ll. 3-20; col. 3, ll. 21-44; l. 8; col. 6, l. 49 to col. 7, l. 19; col. 7, ll. 38-41 and col. 8, l. 25 to col. 12, l. 19), as the SyncML specification is able to work any mobile device, the code would be re-usable for different electronic device.

43. As per claim 40, Rao teaches a server (Fig. 1, ref. 109), comprising:

a memory configure to store commands, execution of which resulting in creation at an electronic device (Fig. 1, ref. 107), of a hierarchical nodular data structure (e.g. tree data structure), having leaf nodes and interior nodes, that comprises first data stored at a first leaf node, and configured to store a further command identifying the first leaf node for operating on the first data stored at the first leaf node (Fig. 1; col. 3, ll. 21-44; col. 4, l. 62 to col. 5, l. 8; col. 5, l. 23 to col. 6, l. 4; col. 6, l. 49 to col. 7, l. 19; col. 7, ll. 38-41 and col. 8, l. 25 to col. 12, l. 19), as the tree data structure includes the leaf nodes and interior nodes and the operation includes downloading and updating processes of update firmware data; and

a transmitter configured to transmit the stored instructions (Fig. 1; col. 3, ll. 21-44; col. 4, l. 62 to col. 5, l. 8 and col. 8, l. 25 to col. 12, l. 19), as the transmitter is needed in order to transfer the command to the mobile handset (Fig. 1, ref. 109).

Rao does not teach the server comprising specifying execution of an unidentified executable on the first data.

Szeto teach a system and method comprising specifying execution of an unidentified executable (e.g. supporting application) on a first data (Fig. 12A and col. 12, l. 66 to col. 13, l. 16), as the specification of the supporting application is in accordance with the property of the first data; more specifically, the received message do not specify the supporting application to be utilized with the message, and after the application type is determined from the received message, the corresponding supporting application is specified.

It would have been obvious for one of ordinary skill in this art, at the time of invention was made to include Szeto's specification of the unidentified executable into Rao's operation of the first data for the benefit of having a reliable system and method for a user to execute and control application (Szeto, col. 2, ll. 30-33) to obtain the invention as specified in claim 40.

IV. CLOSING COMMENTS

Conclusion

a. STATUS OF CLAIMS IN THE APPLICATION

The following is a summary of the treatment and status of all claims in the application as recommended by **M.P.E.P. 707.07(i)**:

a(1) CLAIMS REJECTED IN THE APPLICATION

THIS ACTION IS MADE FINAL. Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

b. DIRECTION OF FUTURE CORRESPONDENCES

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Chun-Kuan (Mike) Lee whose telephone number is (571) 272-0671. The examiner can normally be reached on 8AM to 5PM.

IMPORTANT NOTE

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Alford Kindred can be reached on (571) 272-4037. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/C.K.L./

August 22, 2008

Chun-Kuan (Mike) Lee
Examiner
Art Unit 2181

/Alford W. Kindred/

Supervisory Patent Examiner, Art Unit 2181